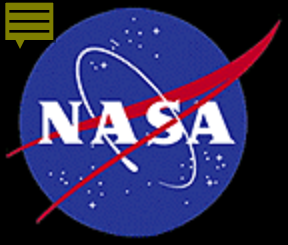


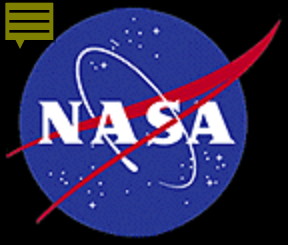
# Using class, activity, and state charts to validate system requirements : *the basics*

John “Mac” Felsing



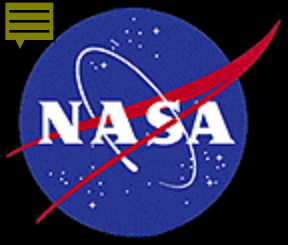
# Objectives

- Show how multiple sources of information are linked and synthesized into an information repository (model).
- Demonstrate how the viewing the synthesized information in a graphical format provides a more powerful analysis perspective
- Show how the information helps to identify complex risks, issues, mismatches, missing information, and inconsistencies that can jeopardize project success.
- Lays the ground work for developing both static and dynamic analysis tools IV&V can use to produce products for our customers.



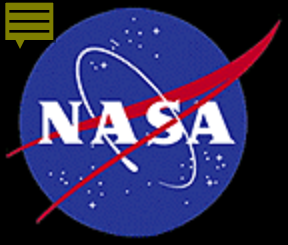
# Why We Model

- Creating models helps us to understand the system under consideration.
- It helps us with the 2 C's (Complete and Consistent)
  - This is a byproduct of the modeling process itself
- It exposes a clearer view of the relationships between the many requirements, specifications, elements, systems, subsystems components, behaviors and interactions.
- It exposes disconnects, missing elements, conflicts, etc. very early on in the programs life cycle.
- The visual presentation format allows us to present different views and perspectives of complex systems of systems and gives us a better understanding of those systems and their interactions.



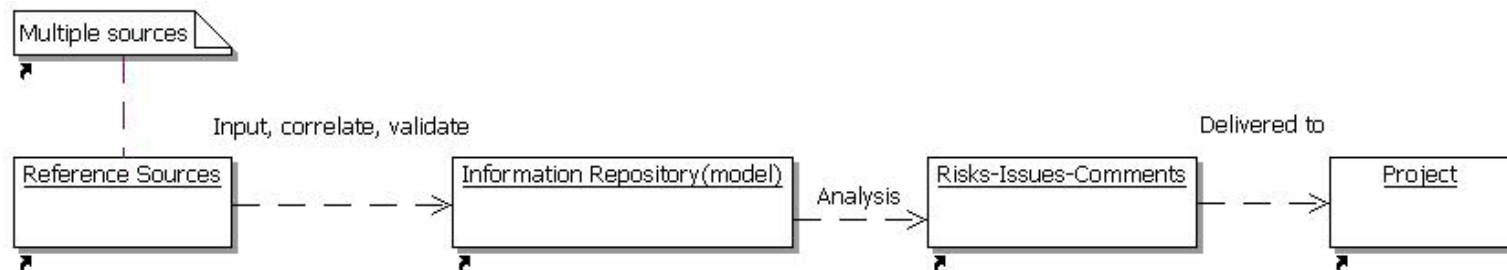
# What modeling produces

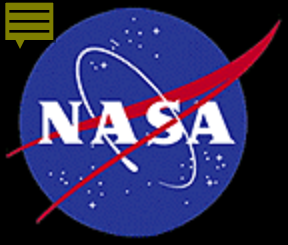
- A **logical information repository** that can be mined and analyzed to answer the 3 questions.
- At IV&V models are NOT our end product.
  - We generate questions which may turn into risks, issues and comments to be given back to the projects to identify and prevent possible catastrophic failures, Loss of Life, Loss of Mission, and Loss of Assets.
- The model is a **tool** to used to help us provide our IV&V products
- Uncover and understand additional scenarios
  - Quickly generate obvious scenarios and explore scenarios that may not be well understood at any one point during the project.
- Provides analysis evidence
- Create various reports about the content/consistency of the information available



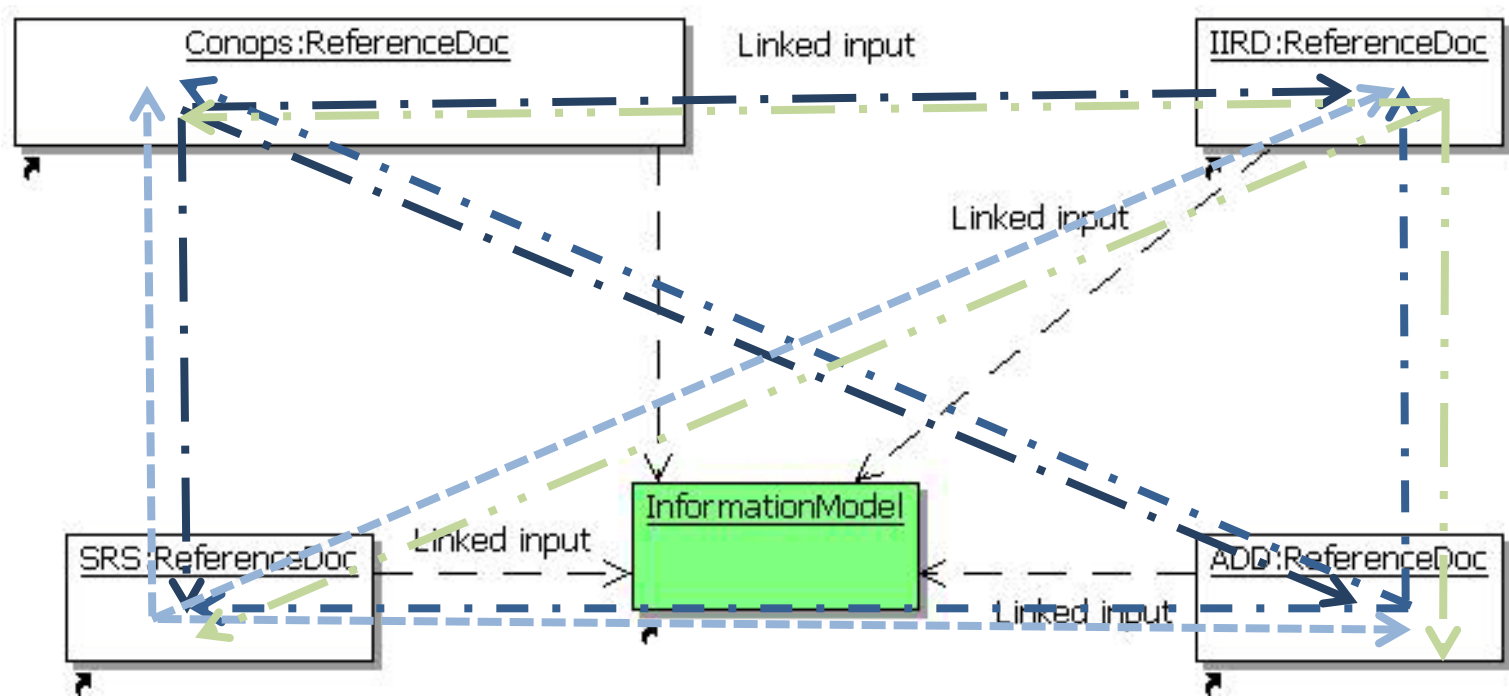
# Information Flow

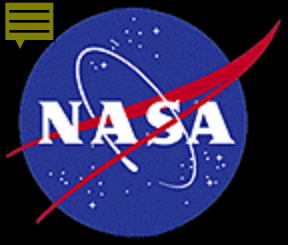
- Sources – (multiple)
- Flow into the Model (Information Repository)
- Output from the model
  - Risks, Issues, Comments
- Delivered to the project





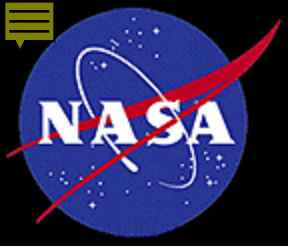
# Multiple Sources





# What information are we hunting?

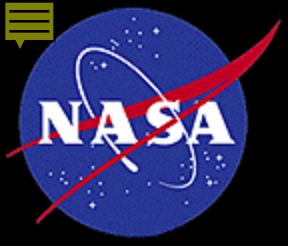
- We look for information (requirements, operational behaviors, interactions, etc) that are:
  - Orphaned
  - Missing
  - Conflicting
  - Ambiguous
  - Incomplete
  - Out of place



# Setting up the information – the Basics

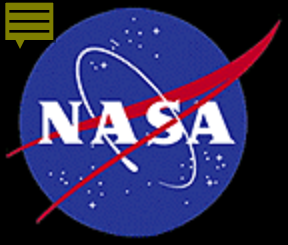
- Classes/Objects
- Activities/Behaviors
- States / Transitions





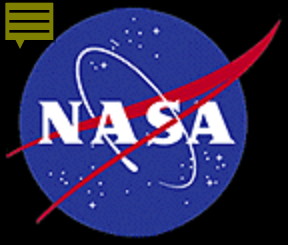
# Classes / Objects

- **Class** – pattern or blueprint of a related set of real world objects.
  - Self aware
  - Behaviors (Methods)
  - Attributes
  - Links to other objects.
- **Objects**
  - An object is something that occurs in the real world.
  - The instantiation of a class.
  - A Software Object is something in a system that models a real world object. The object may be physical, or logical



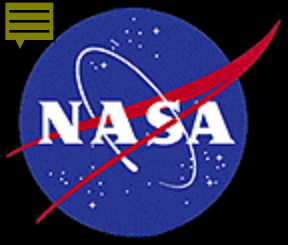
# Activities / Behaviors

- Activities or Behaviors
  - A sequence of labeled events or actions
  - Include steps necessary to accomplish the behavior
  - Pre and Post Conditions
  - Guards and Constraints
  - Nominal and off Nominal flows
  - Data stores
  - Actors (Objects in the form of swim lanes)
  - Interactions between actors (interface)



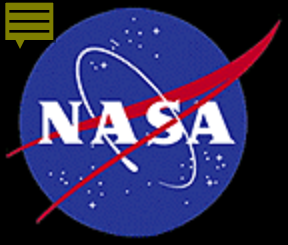
# States and Transitions

- State – value of the attributes of an instantiated class at a given point in time
- Transition – the change in values of the attributes of an object
- Identifies the known states of an object
- Identify the actions that cause transitions between known states



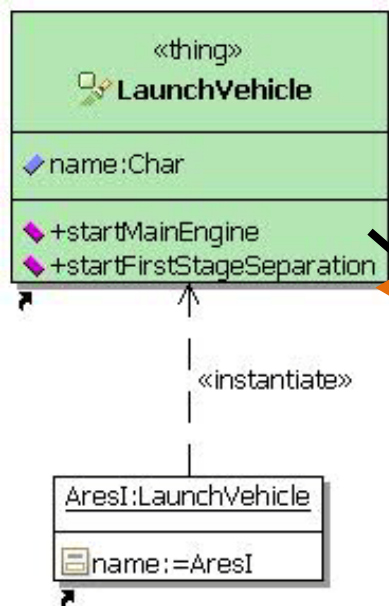
# How we combine the information

- Transitions on a state diagram link to action steps in and activity diagram
- Partitions link to classes and objects
- Activities link to behaviors
- They are all views of the same information

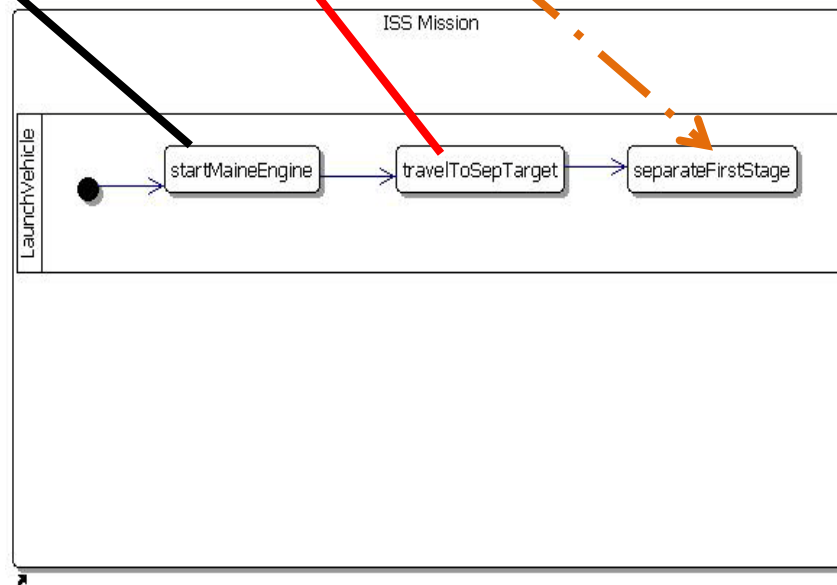


# Linked Information

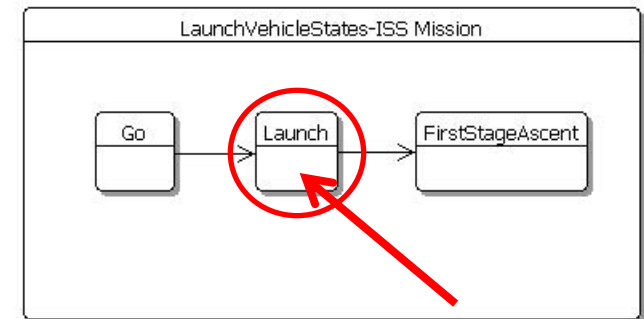
Class



Activity

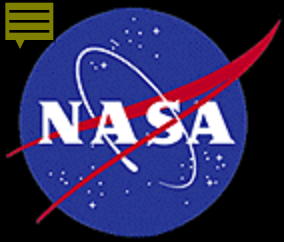


State



Name mismatch?

How do we get to this state?



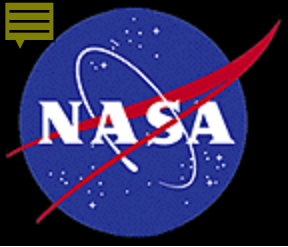
# Orion CEV Software

Packages

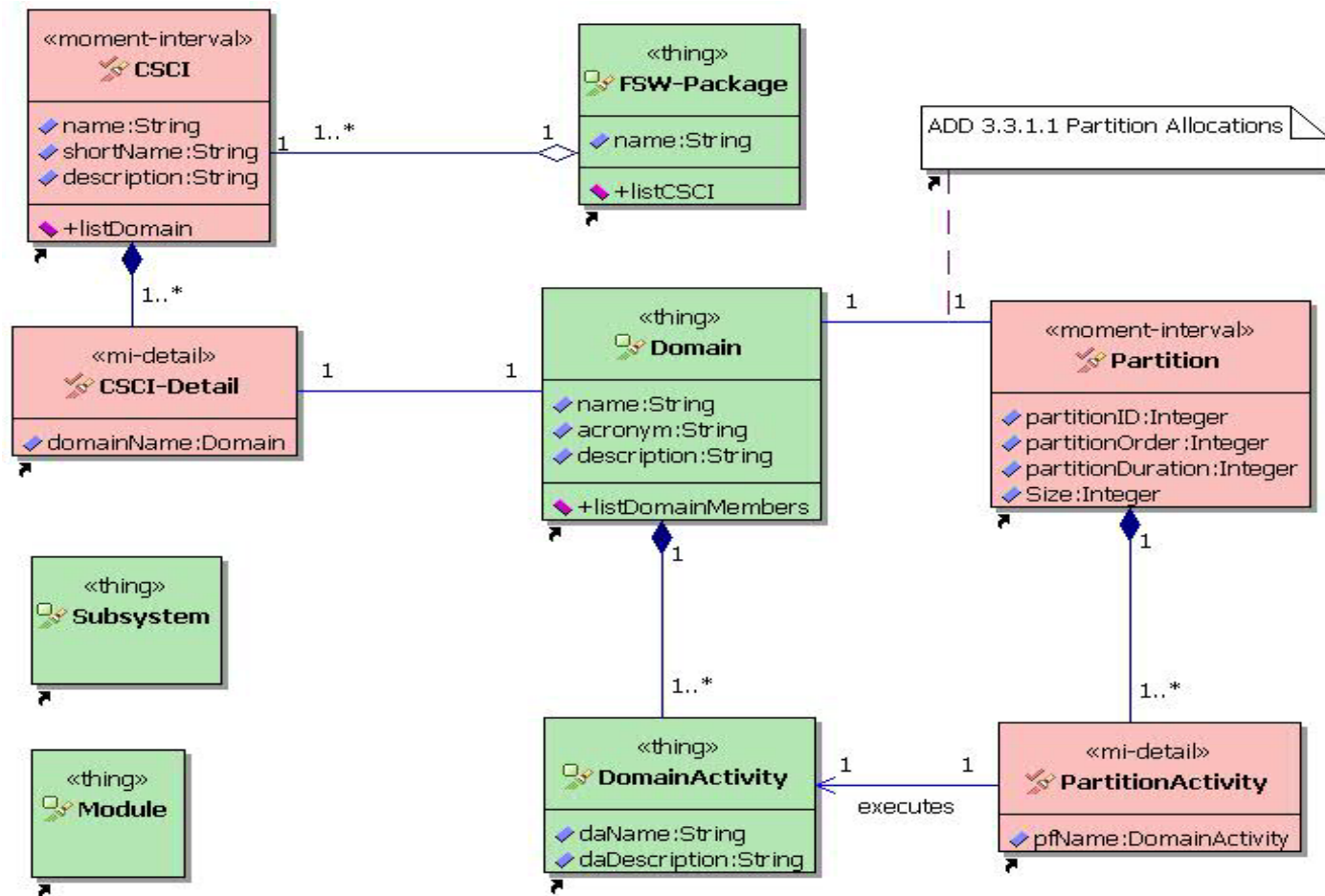
acronym

CSCI

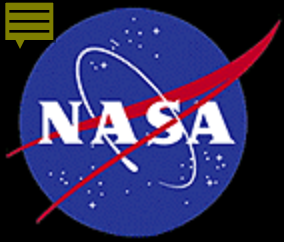
Domain



# Orion Software Organization Structure







# Tracing from Reference to Object

Modeling - CEV Software CSCI Component Structure - Together

File Edit Source Refactor Navigate Search Project Model Diagram Run Window Help

Model Navigator Navigator VM Sequence And Cont VM Configure Vehicle CEV Software CSCI Co

CEV-T-035200 Avionics IRD 824  
CEV-T-047006-Vol\_VII\_ADD 866  
CEV-T-048001 ADL SRS 741  
CEV-T-048021 TMG SRS 869  
3.1.3 Phase and Segment Sequencing 872  
CEV-T-048021 TMG SRS Document Number: CEV  
FSW.03975 Return Validation Response 872  
FSW.03976 Validate Command Source 872  
FSW.03977 Validate Command Content 872  
FSW.03978 Inhibit Capability 872  
[Note Link: [null] -> [Abort Decision]]  
[Note Link: [null] -> [Application Services]]  
[Note Link: [null] -> [Bite Monitor Partition]]  
[Note Link: [null] -> [Caution and Warning Logic]]  
[Note Link: [null] -> [Clock Management]]  
[Note Link: [null] -> [CnDH Subsystem Manage]]  
[Note Link: [null] -> [Command Interface]]  
[Note Link: [null] -> [Command Processing]]  
[Note Link: [null] -> [Command Sequence Auto]]  
[Note Link: [null] -> [Data Recording]]  
[Note Link: [null] -> [EPS Subsystem Health Ma]]  
[Note Link: [null] -> [Event Plan Management]]  
[Note Link: [null] -> [File System Management]]  
[Note Link: [null] -> [Health and Status Message]]  
[Note Link: [null] -> [High Gain Control]]  
[Note Link: [null] -> [Micro MIU Management]]  
[Note Link: [null] -> [Note1]]  
[Note Link: [null] -> [Phase Segment Sequenc]]  
[Note Link: [null] -> [SMMC Executive]]  
[Note Link: [null] -> [Vehicle Health Determina]]  
FSW.03979 Enable Capability 872  
FSW.03980 Execute an Inhibited Function 872  
FSW.03981 Maintain Function State Through Rese  
FSW.03982 Format Command Responses 872  
FSW.03989 Arm Command 872  
FSW.05699 Select Event Plan 872  
FSW.05700 Load Command Validation Parameters  
FSW.05701 Load Inhibit-Enable Transition Data 87  
FSW.09994 Initialize Event Plan 872  
FSW.09996 Initialize Command Validation Paramet  
FSW.09997 Initialize Inhibit-Enable Transition ... 87  
FSW.10005 Evaluate Event Condition 872

Component Structure Diagram:

- Vehicle System Manager (VSM) -> Time and Event Monitor (TEM) -> Time and Event Monitor (TEM) -> Time and Event Monitor (TEM)
- Vehicle System Manager (VSM) -> Basic Event Detection Logic (BEDL) -> Basic Event Detection Logic (BEDL) -> Basic Event Detection Logic (BEDL)
- Vehicle System Manager (VSM) -> Vehicle Health Determination (VHD) -> Vehicle Health Determination (VHD) -> Vehicle Health Determination (VHD)
- Vehicle System Manager (VSM) -> Caution and Warning Logic (CWL) -> Caution and Warning Logic (CWL) -> Caution and Warning Logic (CWL)
- Vehicle System Manager (VSM) -> TMS - All Sampling Ports Monitor (TMS-ASPM) -> TMS-ASPM -> TMS-ASPM
- Vehicle System Manager (VSM) -> SST - System and Subsystem Te On-Bus Check (SST-SSC) -> SST-SSC -> SST-SSC
- Vehicle System Manager (VSM) -> Abort Decision Logic (ADL) -> Abort Decision Logic (ADL) -> Abort Decision Logic (ADL)
- Vehicle System Manager (VSM) -> On-Bus Check (OBC) -> On-Bus Check (OBC) -> On-Bus Check (OBC)

Properties Panel:

Time and Event Monitor selected

Property	Value
instantiated	Domain
specification	Time and Event Monitor
name	Structure.Orion.domain.software.CEV Software.VEHI...
full name	
stereotype	
visibility	Private
metaclass	Instance Specification

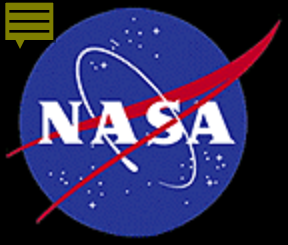
Choose Reference Element to select

Select the desired element.

Elements:

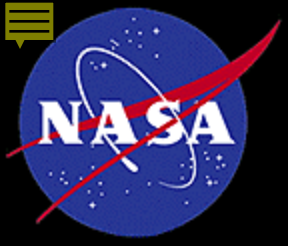
- The TEM (Time and Event Manager)
- 3.2.7.5.3 Time and Event Monitor Di
- FSW.10005 Evaluate Event Condi





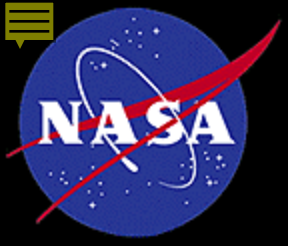
# Ask questions, use model to answer

- Reports on issues
  - References
  - Observations
  - Orphaned elements
  - Missing elements
  - Conflicting elements



# Summary

- Model is an **Information Repository**
  - Allows a more omniscient view of system
  - Find complex inconsistencies and issues among multiple sources at one time
  - Identify areas of ambiguity quickly
  - Justify Issues and discrepancies, their importance and criticality



# Next Steps

- Animation/Dynamic Behavior
  - Validate that a given activity and states support the specified behavior through animation/simulation
- More transforms to find more inconsistencies and issues.
  - Generate classes, interfaces, and component diagrams from activity and state diagrams.
- Automate generation and identification of issues through audits and metrics
  - Use model audits and metrics to quickly identify discrepancies and missing or conflicting information at the diagram level.
- Impact analysis
  - Identify the impact of changing requirements, design and architecture using the information in the model to quickly identify areas that are affected by changes in reference documentation and requirements
- Other
  - Create report templates and queries to quickly extract information for IV&V products, such as issues, observations, missing or conflicting requirements, analysis views, etc.